Problem-Based Learning Applied to Programming Instruction

D. E. Stevenson
Computer Science Department
Clemson University
Clemson, SC 29634-0974
steve@cs.clemson.edu

Abstract

Problem-based learning is an old idea whose time has come again. In PBL, the student is the central focus. The instructor serves as a guide, not a fount of knowledge. This paper focuses on how one person uses the PBL philosophy in computer science education.

Problem-based learning harkens back to the much-romanticized "little schoolhouse on the prairie." In the 19th Century Midwest, the schools had but one teacher to teach all grades. To accomplish this, the teacher used groups to teach one another; *i. e.,* vertically and horizontally integrated groups with the older children helping the younger. The literature on PBL is enormous. Virtually every comparison of PBL with "kill-and-drill" or lecture shows the superiority of PBL.

If it is superior then why not apply it to computer science education?

1 Moving towards Change

Several years ago, I took stock of my teaching techniques and the results I was getting. I was not particularly pleased. Some observations were

1. Students do not productively read code when they are learning to code.
2. Students cannot symbolically manipulate nor abstract code. Students do not work through textbook details of examples.
3. Students cannot develop tests and carry through a test plan.
4. The correlation between scores on homework and final grade is too strong to deny.
5. Unexecuted code written on tests is meaningless.

My conclusions were (1) lectures do not work due to student passivity and (2) active learning is needed to cement concepts. One approach, used in medical schools, was exactly what I was looking for. This approach is called *problem-based learning* (PBL).

I report my use of problem-based learning activities in three courses: a beginning service course, the data structures course, and compiler courses. I focus on programming courses, although I use many of the same techniques in theoretical classes. The remainder of the paper discusses the essentials of PBL (Section 2), how I use these concepts in my classes in Section 3, and some conclusions (Section 4) of two years of strictly PBL teaching practices.

2 Problem-Based Learning Essentials

This discussion is adapted from the discussion of the same name at the University of Southern Illinois School of Medicine[1]. Problem-based learning is the standard pedagogical approach in medical schools, world-wide. Problem-based learning began at McMaster University Medical School in 1969. Their rationale is (1) Students must become life-long, self-motivated learners; (2) Learning takes place in small group tutorials and is problem-based and self-directed; and (3) Progress is monitored by continuous personal and group evaluation. For searching purposes, the following terms are employed for PBL: student-centered, problem-based, inquiry-based, integrated, collaborative, and reiterative learning. The Southern Illinois University School of Medicine, another PBL pioneer, describes

[1] www.pbli.org/pbl/pbl_essentials.htm.

PBL as both a curriculum and a process. The curriculum consists of carefully selected and designed problems that demand from the learner acquisition of critical knowledge, problem solving proficiency, self-directed learning strategies, and team participation skills.

The essential concepts as documented on the Southern Illinois web site are

1. Students must take the responsibility for their own learning. Students must be able to efficiently and effectively identify what they know, what they need to learn, and what resources they need to accomplish their task.

2. Problem-based learning is student-centered. The teacher provides only the problems and experiences that challenge the students to learn what is expected in the curriculum. The teacher guides students with the problem as the students develop problem-solving skills, identify what they need to learn, and develop self-directed learning skills.

3. The problems used in problem-based learning must be ill-structured and allow for free inquiry. The activities carried out in problem-based learning must be those valued in the professional world.

4. Collaboration is an essential skill that students must master.

5. What students learn during their self-directed learning must be applied to the problem with reanalysis and resolution. They must do this in a way that will provide a deeper understanding of the problem and insure the recall of that information when they are faced with similar problems in the future

6. Students must reflect on what has been learned. A closing analysis of the concepts and principles used to solve the problem is essential.

7. Self and peer assessment should be carried out at the completion of each problem and at the end of every curricular unit. The ability to accurately monitor the adequacy of personal performance is essential to developing life-long self-directed study skills. The ability to provide colleagues with accurate feedback is an important skill in life and career.

8. Although a major component of the assessment of students' progress comes from self and peer assessment that occurs at the end of every problem, additional formal assessments must assess the students' problem-solving skills, self-directed learning skills, and ability to recall and apply an integrated knowledge base in work with a problem.

Experts are not rule followers and they are very intuitive about their subjects. Students must work towards this mastery; this expertise is gained by working on problems with the tools the student will ultimately use. For this reason, I long ago adopted the following grading policy: (1) If one can only do something but not explain how: C; (2) If one can only tell me how and not explain why: B; and (3) If one can do it, explain how, and explain why: A.

3 How I Run Class

Robert Lee Moore (1882-1974) was a famous mathematician but he might be even more famous for his teaching. Moore was an early advocate of the inquiry-based/problem-based teaching methods. The archetypal story is worth telling here.

Moore would come into class on the first day as ask, "Are there any questions?" There never were, so he would dismiss the class. This continued until people started asking questions. By then, the uninterested students and those only interested in a grade would have dropped out. Even starting late in the semester, Moore would cover all of the material using his reversed Socratic method.

I am not as confident as Moore. I will ask for questions and cover whatever is asked before moving on to new material. If there are no questions, I try to generate them by challenging the class or giving them an in-class assignment.

- Develop Historically

  You, as an instructor of a course, know the subject, but it is all new to the students. Recreating the context that surrounded seminal problem is a "cheap" way to find PBL exercises.

- Learning Styles

  An underlying message of PBL is "know yourself." One important new tool is that of "learning skills." Of the many inventories I have tried, the "Learning Skills Inventory" developed by Felder has consistently been the most insightful for students. The purpose of the site is to (1) assess the learning style of the student and (2) present strategies that the students can employ when their professor has a different style. Felder uses four axes: global-sequential; active-reflective; visual-verbal; and sensing-intuitive.

  For the complete survey and explanations see

http://www2.ncsu.edu/unity/lockers/users/f/felder/public/ILSpage.html

To effectively use these tools, you must first take the survey. You may be shocked as to your own learning styles. The Felder site has several pages that explain the various learning styles. Next, have the students take the survey. I always announce my own learning styles.

- Lecturing Lecturing is not the first resort, it is the last resort! Lecturing harks back from the days when books were not readily available; this is hardly the case today. I do not lecture, *per se*, but I do come to class with prepared remarks to drive discussions.

- Questions Use student questions to drive class discussion and homework. The data structures course has been taught this way for several years. On the other hand, I have a standing set of problems that is announced at the beginning of the semester which guarantees that the subject is systematically covered.

- Reverse Socratic The Socratic method is generally taken as the teacher asking questions of the students. Moore's procedure is to get the students to ask the questions. I want to help the students reason to a conclusion. Yes, poor students hate this approach because I do not "give them the answer."

- In-class Exercises

  I strive for almost constant chaos in class by having the students doing some exercise together. Planning for this is not as hard as one might think: (1) Look at your old lectures for the topic. (2) What is the point? (3) How can you make the students discover this point rather than tell them? What thought experiment can you devise to make it happen? What short programming project will make them discover what you want?

  Or turn student questions, as they occur, into overnight homework problem. Make them work on it. For example, I rarely answer syntax questions in class I assign it as an overnight problem to be explored.

  Another source of problems in class are the students themselves. For example, they all proudly tell you they have a 600 MHz computer — but they do not usually know what that means. A little discovery here goes a long way to understanding the "Big Oh" notation, for example.

- Standing Problem Sets

  The mechanism to insure that all topics are covered by semester's end is the "standing problem set." This is a set of problems developed before the beginning of the semester; *i. e.*, instead of planning lectures, I plan problems. These problems are chosen to become incrementally more difficult and more complete. The last project or two incorporate most of the material of the course. A great source for these problems is the ACM Repository of problems used in programming contests.

  This, of course, must be different for the different levels of courses. In the freshman class, the standing problems are very simple to state and may even contain specific directions concerning the method of solution. The data structure problems emphasize performance and correctness; the problem statements give much less specific directions. In the compiler class, the problems are taken from the development requirements. This is easy to do because I give the students a Microsoft `Project` file that has the course laid out as a Gannt chart.

- The Use of Teams

  Non-academic organizations survive using teams. Using teams correctly in academia is difficult. In many of my colleagues' mind, students in teams are cheating. I avoid this by making the team component worth one letter grade (ten percent). Every student does every exercise individually. A bit of planning solves the dishonesty problem.

  Students must, however, learn to work with one another. Students need to be supported and the need to learn how to find that support. Read *Talking about Leaving* [10]. The team is used for support. I form base teams by randomly assigning students to three-person teams. I expect these teams to study together and help one another. I use in-class teams to tackle the in-class questions. These in-class teams may be the base team, but I may reassign teams for the day.

- The Lab Report

  Each program is submitted with a laboratory style report. The purpose of the lab report is to get the students to start explaining their actions. The report follows a modified Polya scheme.

  1. The first section is either a user's manual if the exercise is the development of a program/object that someone else would use, or a

discussion of the problem. The students must justify their course of action.

2. Test plan.
3. Source for the program.
4. Test execution captured by some automatic process. Cheaters usually aren't good enough to confute `script`.
5. A section describing what the student learned during the solving of the problem: what new technique, etc, was used and how this helped.

This last point is the most crucial, in my experience. Poor students will slap down something trying to guess what you want to hear. In a recent semester, just the opposite happened: a student had the "light" come on and he wrote about it. He turned in the best projects the rest of the semester.

- Grading

I demand excellence. For prepared homework, my policy is that if it is not according to specifications, I will not grade it until it gets is. This is what students should expect to have happen on the outside.

My grading policy is very straightforward.

1. Running, untested, undocumented program is worth fifty percent.
2. Loss of 10% with permission to resubmit except for egregious offenses is awarded to the following offenses: (1) Late; (2) Failure to following specifications and other software engineering violations; and (3) Failure to test adequately.

There is one problem I have not completely cured. In semester long project courses (compilers) the students want "mid-term grades." For the most part, they have had only one or two milestones by that time and having a reasonable assessment is hard. Long projects need to be carefully planned out. I use Microsoft `Project`. The students are required to use `Project` as a history keeping vehicle. The final report requires that they submit their actual elapsed times. `Project` contains all the milestones and the initial plan.

4 Conclusions

I have presented problem-based learning as I use it in undergraduate and post-graduate classes. Problem-based learning has the students work on problems rather than to have the professor stand up and lecture.

My experiences are

1. Once they get used to it, most students seem to like the give and take of class.
2. Students are much more likely to engage me outside of class, sometimes finding myself in the "in locus parentis" role.
3. Those who got to college by memorization obviously do not like this sort of class.
4. Students self-select for final grade. I have not been surprised.
5. I do not write hourly tests; the homework and other problems serve better. In some quarters, this is called *performance based assessment.*
6. One must guard against "gaming" by the students.

In conclusion, *answer almost any questions with a proviso.* If the question is one that the student should be able to reason to an answer, help them do that reasoning by guiding the discussion. If this question can be reworked into a simple experiment, by all means assign it. If all else fails, assign it as a research project.

References

[1] American College Personnel Association. The student learning imperative. American College Personnel Association, Washington, DC, 1994.

[2] David Boud and Grahame Feletti. *The Challenge of Problem-Based Learning.* Kogan Page Ltd, 1998.

[3] Ernest L. Boyer. *Scholarship Reconsidered: Priorities of the Professorate.* The Carnegie Foundation, 1997.

[4] M. T. H. Chi, M. Bassok, M. Lewis, P.Reimann, and R. Glasser. Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13:145–182, 1989.

[5] M. T. H. Chi, P. J. Feltovich, and R. Glaser. Categorization and representation of physics problems by experts and novices. *Cognitive Science*, pages 121–152, 1981.

[6] Robert Delisle. *How-to Use Problem-Based Learning in the Classroom.* Assn for Supervision & Curriculum Development, 1997.

[7] Rick Gordon. *Problem Based Service Learning : A Fieldguide for Making a Difference in Higher Education.* Education by Design, 2000.

[8] Lewis J. Perelman. *Hyperlearning, the New Technolody and The End of Education.* Avon Books, 1992.

[9] J. M. Savoie and A. S. Hughes. Problem-based learning as classroom solution. *Educational Leadership*, 52(3):54–57, 1994.

[10] Elaine Seymour and Nancy M. Hewitt. *Talking About Leaving: Why Undergraduates Leave the Sciences.* Westview Press, Boulder, CO, 1996.

[11] Luann Wilkerson and Wim H. Gijselaers, editors. *Bringing Problem-Based Learning to Higher Education : Theory and Practice (New Directions for Teaching and Learning No. 68).* Jossey-Bass, 1996.